

HIGH RESOLUTION DYNAMICS LIMB SOUNDER

Originator: C. Craig

Date: 2011-09-30

Subject/Title:

SWATH Module

Description/Summary/Contents

This module performs the read/writes for HIRDLS swath HDF-EOS5 data files.

SWATH – “Swath routines” – Routines to handle reading/writing of HDF-EOS5 Swath formatted files. Uses definition files to drive the specific fields which are read/written. Also can be driven by PROFILEIDs.

Keywords:

Purpose of this Document:

**Oxford University
Atmospheric, Oceanic & Planetary Physics
Parks Road
OXFORD OX1 3PU
United Kingdom**

**University of Colorado, Boulder Center
for Limb Atmospheric Sounding
3450 Mitchell Lane, Bldg. FL-0
Boulder, CO 80301**

EOS

SWATH Module

This module performs the read/writes for HIRDLS swath HDF-EOS5 data files.

It is composed of:

- a structure (SWATH_info) which contains all of the HDF-EOS information about the file
- Read, write, open, close and other supporting routines

Reading/writing is controlled dynamically by the definition files. To add a field, simply add another line to the appropriate definition file and then load the data as another element in the output array.

IMPORTANT: *Reading/writing is done using ProfileID – this may or may not be the same as the index number. The routines look into the ProfileID array to find the requested ProfileID and then the index is returned.*

If modifying the routines, pay particular attention to which HDF_EOS integer values are shorts and longs. Obscure errors can occur from having them incorrectly sized.

- | | |
|--|---|
| • SWATH_Init – Initialize the SWATH_info structure | --- SWATH Initializer and Create routines |
| • SWATH_Create | |
| • SWATH_Open – Open HDF-EOS5 Swath and file | --- SWATH Open |
| • SWATH_Close – Close HDF-EOS5 Swath and file | --- <u>SWATH_Close</u> (file may be corrupt if not closed successfully) |
| • SWATH_SetExtent | |
| • SWATH_GetProfileIndex | --- SWATH ProfileID routines |
| • overloaded SWATH_Read routines | |
| • overloaded SWATH_ReadSub routines | |
| • overloaded SWATH_ReadDynam1D | --- SWATH Read Routines |
| • SWATH_ReadDynam2D | |
| • SWATH_ReadHeader | |
| • SWATH_HIRXXXX_ReadStatic | |
| • overloaded SWATH_Write routines | |
| • overloaded SWATH_WriteSub routines | |
| • overloaded SWATH_WriteDynam1D routines | --- SWATH Write Routines |
| • SWATH_WriteDynam2D | |
| • SWATH_WriteHeader | |
| • SWATH_HIRXXXX_WriteStatic | |
| • SWATH_CopyDynam2D | |
| • SWATH_CopyField | --- SWATH Copy routines |
| • SWATH_CopyFile | |
| • SWATH_CopyGlbAttr | |

SWATH_INITIALIZER and Create routines

SWATH_Init

This routine reads the definition file (the file which details the contents of the HDF-EOS5 Swath file). It loads the information into the SWATH_info structure.

Important note: *This routine MUST be called before a swath file is created.*

Open the appropriate definition file based on the HIRDLS filetype

Read in the individual swath variables

Count the number of swath field variables

For each swath field

 Read fieldname, field dimension list, max dimension list, field number type, filetype (whether it is a geolocation 'G' or data 'D' field), title, units, and unique identifier

Close the definition file

Details of definition file

Each of the variables in the definition file is a string (and must be included in single quotes)

Fieldname: The fieldname should adhere to the Aura Guidelines list. Each word in a fieldname is capitalized and spaces are removed (for example TemperaturePrecision). This is the actual name of the field in the HDF-EOS5 file.

Field dimension list and max dimension list: List of dimensions (in Fortran array order) separated by commas. The max dimension list is the same as the dimension list. The names must match the dimension names that are created in the file (during the SWATH_Create step).

Field number type: The HDF-EOS5 string which describes which type of field it is. Examples are HE5T_NATIVE_DOUBLE, HE5T_NATIVE_REAL and HE5T_NATIVE_INTEGER. A list of the possible values can be found in the toolkit under hdfeos5/include/hdfeos5.inc.

Fieldtype: 'G' for geolocation data –this is data which describes the data, not just geolocation data. It includes SolarZenithAngle and other descriptive fields. 'D' for data – this is the actual HIRDLS data and the associated error fields, etc. The field is placed in either the Geolocation or Data sections of the swath in the HDF-EOS5 file based on this value.

Title: A short description of the data – could be used as a title on plots. This is attached to the data field as an attribute.

Units: field which identifies the units. HIRDLS became CF compliant in V6, so vmr now appears in the title and the units are "1". This is attached to the data field as an attribute.

Unique identifier: "HIRDLS-Specific" or "Aura-Shared" depending on whether the field is shared by other Aura teams or is specific to HIRDLS. This is attached to the data field as an attribute. *Only defined for HIRDLS2 and HIRDLS2ALL files*

Excerpt from sample HIRDLS2 definition file

Comments are italicized in parenthesis and do not appear in the file

'HIRDLS'							(Swath name)
'HIRDLS'							(Instrument name)
'L2'							(Process level)
'2.4.19'							(PGE version – in production, no longer used from this location – SIPS supplies this version number now)
'HIRDLS2'							(HIRDLS filetype)
'Pressure'							(VerticalCoordinate)
'Time',	'nTimes',	'nTimes',	'HE5T_NATIVE_DOUBLE',	'G',	'TAI Time',	's',	'Aura-Shared'
'Latitude',	'nTimes',	'nTimes',	'HE5T_NATIVE_REAL',	'G',	'Latitude',	'deg',	'Aura-Shared'
'Longitude',	'nTimes',	'nTimes',	'HE5T_NATIVE_REAL',	'G',	'Longitude',	'deg',	'Aura-Shared'
'Pressure',	'nLevels',	'nLevels',	'HE5T_NATIVE_REAL',	'G',	'Pressure',	'hPa',	'Aura-Shared'
'SecondsInDay',	'nTimes',	'nTimes',	'HE5T_NATIVE_REAL',	'G',	'Seconds in Day',	's',	'Aura-Shared'
'ProfileID',	'nTimes',	'nTimes',	'HE5T_NATIVE_INTEGER',	'G',	'Profile Identification',	'NoUnits',	'HIRDLS-Specific'
'ScanTable',	'nTimes',	'nTimes',	'HE5T_NATIVE_INT16',	'G',	'Scan Table Identifier',	'NoUnits',	'HIRDLS-Specific'
'SolarZenithAngle',	'nTimes',	'nTimes',	'HE5T_NATIVE_REAL',	'G',	'Solar Zenith Angle',	'deg',	'Aura-Shared'
'LocalSolarTime',	'nTimes',	'nTimes',	'HE5T_NATIVE_REAL',	'G',	'Local Solar Time',	'h',	'Aura-Shared'
'SpacecraftAltitude',	'nTimes',	'nTimes',	'HE5T_NATIVE_REAL',	'G',	'Spacecraft Altitude',	'm',	'Aura-Shared'
'Altitude',	'nLevels,nTimes',	'nLevels,nTimes',	'HE5T_NATIVE_REAL',	'G',	'Altitude',	'm',	'Aura-Shared'
'Temperature',	'nLevels,nTimes',	'nLevels,nTimes',	'HE5T_NATIVE_REAL',	'D',	'Temperature',	'K',	'Aura-Shared'
'O3',	'nLevels,nTimes',	'nLevels,nTimes',	'HE5T_NATIVE_REAL',	'D',	'Ozone vmr',	'1',	'Aura-Shared'
'H2O',	'nLevels,nTimes',	'nLevels,nTimes',	'HE5T_NATIVE_REAL',	'D',	'Water Vapor vmr',	'1',	'Aura-Shared'
'ClONO2',	'nLevels,nTimes',	'nLevels,nTimes',	'HE5T_NATIVE_REAL',	'D',	'ClONO2 vmr',	'1',	'Aura-Shared'
'N2O5',	'nLevels,nTimes',	'nLevels,nTimes',	'HE5T_NATIVE_REAL',	'D',	'N2O5 vmr',	'1',	'Aura-Shared'

SWATH_Info structure contents

This Fortran structure contains all of the information regarding an HDF-EOS5 swath file. It must be filled (using `SWATH_Init`) before the `SWATH_Create` routine is called. For reads, it is filled when the `SWATH_Open` routine is called.

type :: Swath_info_type

!-----

! HDF-EOS file identifiers

!-----

character(len=H2_FilNamLen)	:: filename	! filename
integer	:: fileid	! HDF id for file
integer	:: swid	! HDF id for swath
character*40	:: SwathName	! Name of swath
integer,pointer,dimension(:)	:: IndexProfileID	! Profile ID index array (missing values for profile IDs which are missing)

!-----

! File Level Attributes

!-----

character*6	:: InstrumentName	! 'HIRDLS'
character*6	:: ProcessLevel	! 'L2'
character*75	:: PGEVersion	! Processing version
character*8	:: HIRDLSfiletype	! filetype of HIRDLS file
double precision	:: TAI93At0zOfGranule	! TAI at start of day
double precision	:: HeightTOA	! Height of Top of Atmosphere

!-----

! Swath level attributes

!-----

real,pointer,dimension(:)	:: Pressure	! pressure levels
character*40	:: VerticalCoordinate	! 'Pressure', 'Altitude', etc.

!-----

! Dimension information

!-----

character*40,pointer,dimension(:)	:: DimName	! List of Swath dimension names
integer,pointer,dimension(:)	:: DimSize	! List of Swath dimension sizes

!-----

! Field definitions

!-----

integer	:: nfields	! Number of total fields
integer	:: ngfields	! Number of geolocation fields
integer	:: ndfields	! Number of data fields

character*800,pointer,dimension(:)	:: fieldName	! List of Field names
character*800,pointer,dimension(:)	:: FieldDimList	! List of Field dimensions
character*800,pointer,dimension(:)	:: FieldMaxDimList	! List of Field Maximum dimensions
character*800,pointer,dimension(:)	:: FieldNumType	! List of Field number type
character*1,pointer,dimension(:)	:: FieldType	! 'G' for Geolocation field, 'D' for data field
integer,pointer,dimension(:,:))	:: FieldDimSize	! List of field dimensions (set in SWATH_Create and Swath_Open)

!-----

! Attributes for fields

!-----

real,pointer,dimension(:)	:: MissValue	! Missing value for field
character*120,pointer,dimension(:)	:: Title	! Title for field
character*160,pointer,dimension(:)	:: Units	! Units for field
character*80,pointer,dimension(:)	:: UniqDef	! UniqueFieldDefinition for field (indicates Aura-wide uniqueness)

!-----

! Chunk size for chunking fields in HDF file

!-----

integer	:: chunksize	! Chunk size
---------	--------------	--------------

!-----

! Easy access to nLevels and nTimes dimension values

!-----

integer	:: nLevels,nTimes,nChans,nCloudTypes
---------	--------------------------------------

end type Swath_info_type

SWATH_Create

Create the HIRDLS Swath file, set up the data fields (create them with missing data), assign and fill any attributes attached to the field, and create and assign the dimensions for the file.

Important Note: *SWATH_Init* must have been called prior to this routine to load in the file's descriptive information.

Open the file for writing

Create an HDF-EOS Swath within the file

Loop over dimensions

- Create the HDF-EOS dimension and assign it a value

Loop over data fields

- Based on the type of data, set the appropriate missing value

- Separate out the dimensions in the dimension list

- Assign the dimension values to the `SWATH_info%FieldDimSize`

- Set the chunk size (used for compression) Each chunk is compressed separately

- Define the type of compression

- Define the field

- Set the missing attribute

- Set the local attributes

SWATH_Open

Open the HDF-EOS Swath file. The file's descriptive information is loaded into the SWATH_info structure

Open the file

Retrieve the file level attributes

If a requested swath name is provided, open this swath otherwise open the first swath in the file (usually the only one)

Get the file's dimensions

Check for the special dimensions of 'nTimes', 'nLevels', 'nChans', and 'nCloudTypes'. Assign these within the SWATH_info structure. These and all other dimensions get assigned to the generic DimName and DimSize arrays.

Retrieve list of geolocation and data fields within the file and their dimensions

SWATH ProfileID routines

The SWATH routines are all indexed by ProfileID (which may or may not be the same as index number). The ProfileID is a number assigned to every profile within a file (day) back in the L1 processing. As bad profiles are dropped from the processing stream, the ProfileID number remains constant. This allows profiles to be identified with the same ProfileID between processors. Because of this, all of the swath routines use StartProfileID and StopProfileID to indicate the extent of the data to be read in. Values of 0,0 are used to indicate the full time range within the file.

Swath_SetExtent: This routine sets the appropriate start/stop indices (start, stride, count) for all of the dimensions in the requested field. These indices are then used by the HDF-EOS5 read/write routines. If one of the dimensions is time, the time indices are changed to the appropriate indices for the requested StartProfileID/StopProfileID.

SWATH_GetProfileIndex: Retrieves the indices of the StartProfileID and StopProfileID in the time dimension. Also checks for IDs which are outside the start/stop of the array of ProfileIDs and writes an error and exits if necessary.

SWATH Read/Write Routines

These read/write routines are wrappers around the HDF-EOS5 read library calls. They use the ProfileID to index the time dimension. There are two ways of reading /writing the HIRDLS Swath files, either directly or dynamically. Dynamic reading/writing is used when field information contained within the SWATH_info structure is used to loop over the fields. The dynamic fields are just the “data” fields in the SWATH. The geolocation fields cannot be read dynamically. For reading this data comes from the file’s definition file. For writing, the SWATH_info field is initialized with all of the data fields within the file. Looping over fields within the SWATH_info structure allows additional fields to be added to the file without any code changes. The vast majority of the HIRDLS2 files use this dynamic interface. Internally, they use the direct access routines. The direct access routines may be called by the user as well.

Direct access calls:

SWATH_Read/SWATH_ReadSub
SWATH_Write/SWATH_WriteSub

Dynamically access calls:

SWATH_ReadDynam1D/SWATH_ReadDynam2D
SWATH_WriteDynam1D/SWATH_WriteDynam2D/SWATH_WriteDynam2DB

Support routines when reading/writing dynamically:

SWATH_ReadHeader and SWATH_HIRXXXX_ReadStatic
SWATH_WriteHeader and SWATH_HIRXXX_WriteStatic

Direct Access calls

SWATH Read: Overloaded function. Allocates the pointer array to the appropriate dimensions and reads in the requested data.

SWATH ReadXXSub: Where XX= 2b, 2c, 2r, and 3r. These routines are used for a subsection of a pointer of the type indicated. The actual read internally is one dimension less than the call (for example 2r is a subsection of a 2 dimensional real pointer which becomes a 1 dimensional real array in the routine). *Space for the array must be allocated before these routines are called.* Reads in the requested data.

SWATH Write: Overloaded function. Writes out the data to the requested field

SWATH WriteXXSub: where XX=2r, 2b, 2c, 3r, 3b, 4r. These routines are used for a subsection of a pointer of the type indicated. The actual write internally is one dimension less than the call (for example 2r is a subsection of a 2 dimensional real pointer which becomes a 1 dimensional real array in the routine). Writes out the data to the requested field.

Dynamically access calls

SWATH ReadDynam1D: Overloaded function. Loops over all fields requested, reading the data into a 2D array (1D data and the other dimension is for the fields). Allocates the array and reads in the data requested by the fieldnames and StartProfileID/StopProfileID.

SWATH ReadDynam2D: Loops over all fields requested, reading the data into an array where one of the dimensions is over the fields. Allocates the array and reads in the data requested by the fieldnames and StartProfileID/StopProfileID.

SWATH WriteDynam1D: Overloaded function. Loops over all fields requested, writing out the data in the array where one of the dimensions is over the fields. Writes out the data requested by the fieldnames and StartProfileID/StopProfileID.

SWATH WriteDynam2D: Loops over all fields requested, writing out the data in the 3D real array (2D data with another dimension for fields). Writes out the data requested by the fieldnames and StartProfileID/StopProfileID

SWATH WriteDynam2DB: Same as SWATH_WriteDynam2D excepts writes out a one byte integer.

Support routines when reading/writing dynamically

SWATH ReadHeader: Reads in the time, latitude, longitude and ProfileID and determines the index for the ProfileIDs.

SWATH HIRXXX ReadStatic: These routines read all of the other geolocation information in the HDF-EOS5 SWATH file. The fields are all individually read (no dynamic looping is done).

SWATH WriteHeader: Writes out the time, latitude, longitude and ProfileID and determines the index for the ProfileIDs. Also writes out the file and swath level attributes.

SWATH HIRXXX WriteStatic: These routines writes out all of the other geolocation information in the HDF-EOS5 SWATH file. The fields are all individually written (no dynamic looping is done).

SWATH Copy routines

SWATH CopyDynam2D: Allocate space for the output array and the output IndexProfileIDSub array and copy the input arrays to the output.

SWATH CopyField: Copies the data from the input fieldname in the input file to the output fieldname in the output file. This routine can be used to rename an output field (like H2O18 to H2O) as well as just copying the selected field from an input to an output file. Special handling is done to remove the unused dimension from the CloudTopPressure array if it is present. The output data field must already exist in the output file (have been created) before the copy is called.

SWATH CopyFile: Using the fields specified in the output file's definition file, all of the specified fields are copied from the input file to the output file. The file will be created within this subroutine and all of the attributes as well as data are copied over. If the optional fldNameChange variable is passed in, then the data from the input fieldnames are copied to the output fieldnames.

SWATH CopyGlbAttr: Copies all of the global attributes from one SWATH_info structure to another.